Implementing Heterogeneous Agents in Dynamic Enviornments, A Case Study in RoboCupRescue

Jafar Habibi Mazda Ahmadi Ali Nouri
Mayssam Sayyadian Mayssam M. Nevisi

Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
habibi@sharif.edu
(m_ahmadi,nouri,sayadian,nevisi)@ce.sharif.edu

Abstract. Design and construction of multi-agent systems is a challenging but an intriguing problem. It is because of the intrinsic distribution of the intelligent components. In such environments the interaction and communication between the constituent parts extends the complexity since appropriate coordination methods need to be designated and employed. In this paper a successful experiment in designing and implementing such an environment is presented [1]. The test bed for this research is the rescue simulation environment. The architecture of the implemented heterogeneous agents takes advantage of various algorithms. These algorithms make the agents act intelligently by themselves albeit they happen to act quite in coordination with each other. The implemented algorithms for the sake of cooperation between the heterogeneous agents enhance the overall pay off of the system. The autonomy of the agents is guaranteed by means of some methods such as reinforcement learning, decision trees and some sort of heuristic functions. In order to settle the agents in coordination with each other and make them act cooperatively, some other methods have been used. Among these methods, combinatorial auctions, coalition formation, function approximation for evaluating the value of cooperation, and some probabilistic and heuristic methods can be named.

1 Introduction

Nowadays design and construction of multi-agent system infrastructures is a challenging but interesting problem. Designing systems for soccer player robots[9], computer-aided design of a generic robot controller for a multi- robot system[13], design and implementation of automated highway systems[3], and the hot topic of trading agents[12] are a few examples of the works in this field. In this paper we try to present our experiment in designing and implementing a complex environment of multi heterogeneous agents, so that it can be used for further similar activities.

[1] The implemented agents won the first place award in RoboCup Rescue Simulation league in Fukuoka, Japan, 2002.

The engaged test bed is the rescue simulation environment. In this environment the goal is to rescue a disastered city according to an earthquake. Three kinds of completely different agents are aimed to minimize the overall damage to the city. Such agents have various abilities and hence different responsibilities such as extinguishing fiery buildings, rescuing injured civilians, etc. Also they are supposed to come across a mutual agreement so that their cooperation and coordination would enhance their collaborative efforts and this adds to their complexity. Two aspects of a multi-agent system with intelligence are eligible to note. The first is the intelligence of each agent - the micro level issues. The other is considered with the system as a whole and it is the agents' coordination and cooperation to reach desired goals. In the implemented system both issues are considered and emphasized. It means that although each agent tries to perform his assigned tasks as perfect as possible, he tries to act so that the overall system benefits. In other words the agents are not selfish.

This paper is organized as follows: After introduction we present some general explanation of the rescue simulation environment as our test bed. In section three the fire brigade algorithms are presented. In section four the police agent related topics are discussed. In section five ambulance team related algorithms are presented. Inter-agent communication methods and related subjects are discussed in section six. In section seven a comparison of experimental results is presented and we come to conclusion in that section.

2 Fire Brigade Agents

In this environment, the fire brigades are responsible for controlling the spreading fire in the city, and extinguishing the buildings as many as possible. For this purpose, each agent takes advantage of his visual perception and identifies the buildings on fire. For each fiery building the agent autonomously tries to estimate how dangerous that building is and how much it threatens its neighbors. After this phase, the fire brigades need to act upon the world's situation in a unified approach to increase their coordination. The most obvious approach in this phase is finding the most important fiery buildings and extinguishing them. So, the way agents calculate a building's priority plays an important role in this phase.

A proposed workflow for a fire brigade is shown in figure 1. As the figure suggests, the workflow contains four phases, namely: inception, analysis, decision making, and implementation. This means the agent first receives raw information about the environment. Then by means of communication with other agents, his experiments and his experiences the agent uses this information to gain some kind of knowledge that would be useful in the decision making phase. Then the agent's world model is investigated to find appropriate targets. The most useful and the best estimated targets are selected in this decision making process. In the last phase the agent implements the desired actions according to the target he has chosen.

The decision making section is the most important phase in the agent's workflow in each cycle. The fire brigade agents use a two layered architecture for this

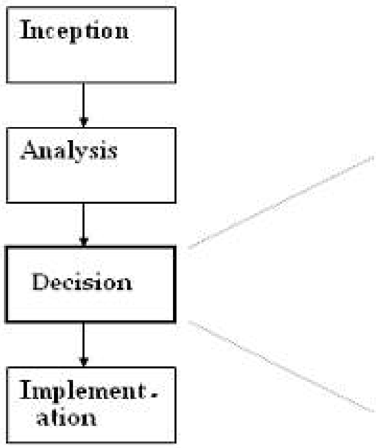

Fig. 1. Agent's workflow

phase. In the first stage, the agents decide independently. They do not care the state of other agents and they selfishly choose some targets for their own. In the second stage, the agents try to coordinate with other fire brigades and also they try cooperating with other heterogeneous agents - such as the ambulance teams and the police forces. This way the overall rescue integrity is guaranteed and this collaboration enriches the result and performance of the overall system. Of the advantages of this architecture is the independence of the two layers. This enables implementation and evaluation of different algorithm in each layer.

2.1 Preliminary Decision Making Based on Agents' Selfishness

The implemented agents in this system estimate the buildings' priorities in an integrated manner. This means they all use the same algorithms for this purpose. In this method each agent assigns an importance number to each fiery building independent of other agents. In this phase the criteria according which the importance of buildings are calculated is highlighted. This is because in this stage the buildings are the targets and agents are assumed to act according to their selected targets. After selecting a few targets from the many available, they are sorted according to their importance. The importance of each building is calculated according to some measurements and criteria that all the agents agree. These measurements are extracted and calculated in the analysis phase. Therefore the way such criteria are extracted and calculated gets very important and determining. The agents in this system calculate the priority of buildings according to their intrinsic importance and their strategic importance.

- *Intrinsic Importance:* The intrinsic importance of each building is a measure of the building's importance in the city by itself and independent of the current situation of the disaster. Hence one factor in calculating the intrinsic importance is the building's geographical situation. Another factor for each building is related to the number of the neighboring buildings that are possible to get on fire if the building gets ignited.

- *Strategic Importance:* The strategic situation of a building is dependant on the way all the buildings are on fire in the city and how the sites of fire are distributed in the city[2]. It is known that the best strategy is finding sites of fire and make them stop spreading by extinguishing the buildings on their boundaries. Hence for each building, it would increase its important if it was located on the boundary. Also the importance of the site of in which the building is located is very important. These criteria altogether conclude the strategic importance of a building.

After each agent finds a set of important buildings independent of other agents, he considers this set of buildings as his potential targets. The agents regard these targets as the input to the next layer of decision making phase. In the next stage the targets are manipulated so that they will be assigned to agents distributedly and in a way that the cooperation and coordination between them is guaranteed.

2.2 Improvement in Decision Making, Seeking Coordination and Cooperation

When designing fire brigades it is really important to note the significance of the coordination between the homogeneous agents. That is because buildings can be extinguished much faster if multiple fire brigades are assigned to a fiery building. This fact has a deep effect on the performance of the fire brigades. Therefore our agents tend to form coalitions with other fire brigades to extinguish fiery buildings with each other. Each coalition is a set of agents with the same target. The coalition members try to reach their goal and put out the building of their choice collaboratively. Since there are too many targets (buildings) and too many agent combinations, the number of possible coalition structures increases drastically. It is required to find an optimal coalition structure between so many possible coalition structures. For this purpose, the agents in this system are regarded as resources that will be assigned to on fire buildings. It means to each fiery building zero, one, two, or more fire brigades can be assigned. Some researches on collaborative planning for resource allocation have been done previously and useful algorithms have been developed. From this approach this problem can be viewed from a combinatorial auction perspective.

The combinatorial auction problem has been emphasized in research recently mostly in combinatorics. In this problem agents are allowed to bid directly for bundles of resources. Suggesting a set of resources, the agents propose a value for them. Then the auctioneer is assumed to allocate resources to bidders so that the overall benefit for the system is maximized.

In the proposed system, the auctioneer is simulated so that the optimal resource allocation and coalition formation is achieved. In this system the buildings

[2] A group of connected fiery buildings are called a site of fire.

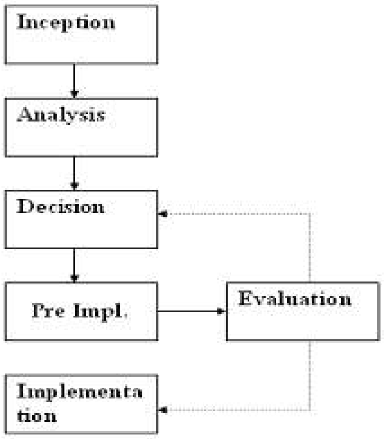

Fig. 2. Agent's workflow with feedback

are assumed to play the role of bidders and agents are regarded as the extinguishing resources. So it can be assumed that buildings suggest values for a bundle of fire brigades. To evaluate each bid the followings are important:

- The importance of the building which is an indicator of its strategic and intrinsic importance.
- The amount of time it takes the proposed coalition to put out the building. This measure can be calculated by means of statistical and computational analysis.

The most complex part of the combinatorial auction is the winner determination algorithm for the auctioneer. As a matter of fact, the most difficult part of the combinatorial auction simulation is to find an optimal coalition structure among the many possible, so that the overall payoff of the system is maximized. Since the algorithms for this problem are NP, in this system the approximate algorithms developed by Edo Zurel have been exploited. This algorithm uses some heuristic functions to prune the search space very fast. Then it uses some greedy methods and by means of hill climbing it finds a very good coalition structure which is an approximate of the optimal solution. This way by simulating a coalition formation via combinatorial auction for the fire brigades the inter-agent coordination is guaranteed in a satisfactory manner and it is very close to the optimal solution.

2.3 More Improvement in Decision Making

The mentioned agent workflow in figure 1 suffers a major drawback: In decision making phase the fire brigades try to select the best possible coalition structures and as to collaboratively extinguish the target buildings. As a matter of fact these coalition structures are formed in the implementation phase. The performance of the system is tightly related to the successful implementation of these coalition formations. According to this workflow, no feedback is given to the decision making phase so it is not strange if a selected coalition structure can not be formed in the next phase. This can be as a result of some problem such as path planning. Hence all the agents that are supposed to form that coalition are failed in their mission. Also it must be noted that the feasibility of path planning can not be regarded as a factor in the decision making. It is because path planning is a time consuming and complex activity. In this system to leverage the impact of this problem the mentioned workflow is changed in two steps.

The first change is in the implementation phase and it is in path planning. The implemented algorithm in path planning uses the same raw data that have been extracted for decision making in the analysis phase. Hence, without direct invocation of path planning algorithm in decision making, it can be guaranteed that a target which will fail to reach in path planning will not be selected in decision making. Also the selected paths using this method are more confident than those extracted by means of classic algorithms and other proposed methods.

The second change is in the architecture of the agent's workflow. Although the strategies mentioned above play an important role in increasing the confidence of finding reachable targets and hence decreasing the possibility of loosing cycles, but still loosing a working cycle could be a huge burden over the system's performance and may result in malfunctioning of agents. Therefore some changes have been made to the agent's workflow. This change is in form of adding a feedback from the implementation layer to the decision making layer (see fig. 2). This way after selecting a target, it would be routed and the result would be evaluated. If the extracted path was applicable, then the decision would be implemented. Otherwise the selected building would be removed from candidate buildings and the agent would redecide. According to the experiments, since the buildings are spread all over the city, usually in the 300 working cycle of the agents, only the agents needed to redecide more than three times for 5-10 cycles. Also they managed to find a well suited target by two times of redecision.

3 Police Force Agents

In Rescue Simulation, the police forces are supposed to clear roads. Trying to clear more roads is not the optimum action. Polices have to select the most important roads. Importance of the road is how many times other agents will pass through this road in the following cycles. Although the exact value of this measure is not computable before the simulation, the probable value is computed. For this purpose, roads are prioritized according to two types of parameters, dependant to the system state, and independate of the current state of the system.

Static value of roads: It is obvious that some roads like those around the city are less important than the central roads. To estimate this kind of importance, it is assumed the roads that are subject to use more frequently are more important. We computer this value in this way:

1. The city is saved in form of graph so that the nodes are the graph's vertexes and the roads are the edges.
2. Taking advantage of Floyd's shortest path, all shortest paths between all pairwise nodes are computed and saved.
3. For each edge (road), the number of occurance of that road in shortest paths are counted.
4. The static value of each road is equal to the number of occurrence of this road in all the shortest paths. This way the road that is more reliable to appear in a selected path is more important.

Dynamic value for roads: At first all dynamic road values are equal to one. During the simulation if an agents requests for clearing a road, its priority is multiplied by ten. The roads around the fiery buildings are also multiplied by two. As time passes this dynamic value for each road is decreased.

Total roads values: It is computed as the road's dynamic value times its static value.

This roads values are considered when the police has decided what to do according to his state in the disaster. In order ro improve the police forces' in decision making, a reinforcement learning has been used. In this method each agent has three actions as follows:

- Stay in his area: When this action is choosed, the police stays in his area and walks around the area he is currently in.
- Help other agents: Selection of this action means the police will be leaving his curent area so as to clear a specific road to help another agent achieve his goal
- Change the area of responsibility

Q-Learning has been used for this learning. In latter version os Arian's police forces an algorithm based on function approximation has been used. More details could be found in [1].

4 Ambulance Teams:

Ambulance team agents rescue injured civilians. They obtain information of civilians by means of communication, and gathering auditorial and visual information. In order to determine whether to go to rescue a civilian or move around to find an injured civilian, a QLearning method has been used.

5 Inter-agent Communication

In the inter-agent communication model, each agent can communicate with its homogeneous agents as well as with its center. In the implemented communication system there are two kind of messages:

- Informative messages: These are the messages which are aimed for informing the agents. For example police forces inform ambulance teams of the injured civilian positions.
- Imperative messages: These messages are requests for help. For example fire brigades ask police forces to clear a specific road.

One of the challenges each agent faces is determining what to do among his own decisions according to his intelligency and the requests recieved from other agents. Design and implementation of a useful method for selecting the best messages to evaluate was of the most important challenges the implemented agents had to face. In the suggested algorithm, each agent sets a priority in the message that is dependent on the following factors :

- Goal's importance: the more important the goal of the sender, the more important is his message.
- The importance of the request in reaching the agent's goal.
- The importance of other possible actions the agent can do if not considering his request.

Each agent with his internal algorithm finds a priority for his decision and comparing this priority with the priority of the messages decides what to do.

Since the way the agents' priorities may be different from each other, and from the priorities set in the messages, the agents use a function approximation method so that in case of failure, the way they set priorities would change.

5.1 Message Evaluation

In our communication system every agent who asks for help, sets a priority in that message. This priority indicates the importance of the situation and how accurate the information is.

One of the problems in communication is to derive a value for the message. we derived an equation for evaluating the message, that is:

$$V_i(t) = f(t) * P(i)/succ(t-1)$$

In the above equation $V_i(t)$ indicates the value of i'th message in each time period and of course each agent has its own queue of messages and has its own $V_i(t)$. $f(t)$ is a coefficient which is related to time and so makes $V_i(t)$ related to time. and so makes $V_i(t)$ related to time. $f(t)$ is what we are aiming to learn and has a constant value at initial state. By learning $f(t)$ we can say that $V_i(t)$ is learned. $P(i)$ indicates the priority that the message has. $succ(t-1)$ is the measure of the success of the agent in last time period. If this value of $f(t)$ is less than a threshold we will accept the message and do what the message has told us to do, if not, we will ignore that message.

We use and incremental function approximation method for learning the value of $f(t)$. In our function approximation method we use an initial value for $f(t)$ and in each cycle we update the $f(t)$ function. The way we update this function has an explicit relation with the agent abilities and the environment of the problem. Further information and more detailed discussion on the above method and the formula can be find in [1] .

6 Experimental Results and Conclusion

The implemented agents has participated in the RoboCup Rescue Simulation league in the RoboCup competitions. For the Seattle-2001 competitions, the Arian team succeeded to won the silver medal. In the Fukuoka-2002 competitions, the Arian team managed to become the world champion of the league. Undoubtedly achieving such positions, specially the goledn medal in 2002 competitions were the result of the methods and algortithms which we discussed in this paper.

The way participating teams are evaluated agains each other is based on the number of dead civilians, and the number of burned buildings. In the following tables a comparison of Arian score and other semi-final and final teams's scores are reflected. It must be mentioned in the competitions the cooperation and coordination between Arian agents were very dominent.

References

1. Mazda Ahmadi, Mayssam Sayyadian, Jafar Habibi, A Learning Method for Evaluating Messages in Multi Agent Systems, Proceedings of the Agent Communicatin Languages (ACL2002) , Italy, Bolognia
2. A.V. Aho, J.E Hopcraft and J.D Ullman, Data Structures and Algorithms, Addison-Wesley, 1983
3. A. Deshpande, D. N. Godbole, A. G. and P. Varaiya, "Design and Evaluation Tools for Automated Highway Systems", Hybrid Systems pp. 138-148, 1995.
4. Jafar Habibi, Mazda Ahmadi, Ali Nouri, Mayssam Sayyadian, Mayssam Mohammadi Nevisi, Utilizing Different Multiagent Methods in RoboCupRescue Simulation, Springer Verlog, 2002
5. Jafar Habibi, Ali Nouri, Mazda Ahmadi , An Approach to Multi-Agent Communication Used in RoboCupRescue, Springer Verlog, 2002
6. Luke Hunsberger, Barbara J. Grosz, A Combinatorial Auction for Collaborative Planning, Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000), pages 151–158, 2000
7. H. Kitano, S. Tadokor, H. Noda, I Matsubara, T. Takhasi, A. Shinjou, and S. Shimada. Robocup-rescue: Search and rescue for large scale disasters as a domain for multi-agent research. In Proc. of the IEEE Conference on Systems, Men, and Cybernetics, 1999.
8. Nair,R., Ito,T., Tambe,M., Marsella,S.:RoboCup-Rescue:A Proposal and Preliminary Experiences. Proceedings of RoboCup-Rescue Workshop, Fourth International Conference on Multi-Agent Systems(ICMAS-2000), July 2000.
9. I. Noda and P. Stone, "The RoboCup Soccer Server and CMUnited Clients: Implemented Infrastructure for MAS Research" Journal of Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publisher, 2002.
10. "http://www.robocup.org"
11. Van Roy, B., Learning and Value Function Approximation in Complex Decision Processes, Ph.D. Thesis, Massachusetts Institute of Technology, May 1998. 30
12. T. Sandholm and V. Lesser, "Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework", Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95), The MIT Press: Cambridge, MA, USA, San Francisco, CA, USA, pp. 328–335, 1995.

13. D. Simon, B. Espiau, E. Castillo and K. Kapellos, "Computer-aided design of a generic robot controller handling reactivity and real-time control issues", RR-1801, 1995.
14. Christopher J. Watkins and Peter Dayan. Q-learning. Machine Learning, 8:279–292, 1992.
15. Edo Zurel, Noam Nisan, An Efficient Approximate Allocation Algorithm for Combinatorial Auctions, Proceedings of ACM Conference on Electronic Commerce (EC-2001), 2001